

Efficient coordination mechanisms for unrelated machine scheduling*

Ioannis Caragiannis[†]

July 12, 2011

Abstract

We present three new coordination mechanisms for scheduling n selfish jobs on m unrelated machines. A coordination mechanism aims to mitigate the impact of selfishness of jobs on the efficiency of schedules by defining a local scheduling policy on each machine. The scheduling policies induce a game among the jobs and each job prefers to be scheduled on a machine so that its completion time is minimum given the assignments of the other jobs. We consider the maximum completion time among all jobs as the measure of the efficiency of schedules. The approximation ratio of a coordination mechanism quantifies the efficiency of pure Nash equilibria (price of anarchy) of the induced game.

Our mechanisms are deterministic, local, and preemptive in the sense that the scheduling policy does not necessarily process the jobs in an uninterrupted way and may introduce some idle time. Our first coordination mechanism has approximation ratio $\Theta(\log m)$ and always guarantees that the induced game has pure Nash equilibria to which the system converges in at most n rounds. This result improves a bound of $O(\log^2 m)$ due to Azar, Jain, and Mirrokni and, similarly to their mechanism, our mechanism uses a global ordering of the jobs according to their distinct IDs. Next we study the intriguing scenario where jobs are anonymous, i.e., they have no IDs. In this case, coordination mechanisms can only distinguish between jobs that have different load characteristics. Our second mechanism handles anonymous jobs and has approximation ratio $O\left(\frac{\log m}{\log \log m}\right)$ although the game induced is not a potential game and, hence, the existence of pure Nash equilibria is not guaranteed by potential function arguments. However, it provides evidence that the known lower bounds for non-preemptive coordination mechanisms could be beaten using preemptive scheduling policies. Our third coordination mechanism also handles anonymous jobs and has a nice “cost-revealing” potential function. We use this potential function in order, not only to prove the existence of equilibria, but also to upper-bound the price of stability of the induced game by $O(\log m)$ and the price of anarchy by $O(\log^2 m)$. Our third coordination mechanism is the first that handles anonymous jobs and simultaneously guarantees that the induced game is a potential game and has bounded price of anarchy. In order to obtain the above bounds, our coordination mechanisms use m as a parameter. Slight variations of these mechanisms in which this information is not necessary achieve approximation ratios of $O(m^\epsilon)$, for any constant $\epsilon > 0$.

*A preliminary version of the results of this paper appeared in *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 815-824, 2009.

[†]Research Academic Computer Technology Institute & Department of Computer Engineering and Informatics, University of Patras, 26500 Rio, Greece. Email: caragian@ceid.upatras.gr.

1 Introduction

We study the classical problem of *unrelated machine scheduling*. In this problem, we have m parallel machines and n independent jobs. Job i induces a (possibly infinite) positive processing time (or load) w_{ij} when processed by machine j . The load of a machine is the total load of the jobs assigned to it. The quality of an assignment of jobs to machines is measured by the makespan (i.e., the maximum) of the machine loads or, alternatively, the maximum completion time among all jobs. The optimization problem of computing an assignment of minimum makespan is a fundamental APX-hard problem, quite well-understood in terms of its offline [37] and online approximability [4, 9].

The approach we follow in this paper is both algorithmic and game-theoretic. We assume that each job is owned by a selfish agent. This gives rise to a *selfish scheduling* setting where each agent aims to minimize the completion time of her job with no regard to the global optimum. Such a selfish behaviour can lead to inefficient schedules from which no agent has an incentive to unilaterally deviate in order to improve the completion time of her job. From the algorithmic point of view, the designer of such a system can define a *coordination mechanism* [14], i.e., a *scheduling policy* within each machine in order to “coordinate” the selfish behaviour of the jobs. Our main objective is to design coordination mechanisms that guarantee that the assignments reached by the selfish agents are *efficient*.

The model. A scheduling policy simply defines the way jobs are scheduled within a machine and can be either *non-preemptive* or *preemptive*. Non-preemptive scheduling policies process jobs uninterruptedly according to some order. Preemptive scheduling policies do not necessarily have this feature and can also introduce some idle time (delay). Although this seems unnecessary at first glance, as we show in this paper, it is a very useful tool in order to guarantee coordination. A coordination mechanism is a set of scheduling policies running on the machines. In the sequel, we use the terms coordination mechanisms and scheduling policies interchangeably.

A coordination mechanism defines (or induces) a game with the job owners as players. Each job has all machines as possible *strategies*. We call an *assignment* (of jobs to machines) or *state* any set of strategies selected by the players, with one strategy per player. Given an assignment of jobs to machines, the cost of a player is the completion time of her job on the machine it has been assigned to; this completion time depends on the scheduling policy on that machine and the characteristics of all jobs assigned to that machine. Assignments in which no player has an incentive to change her strategy in order to decrease her cost given the assignments of the other players are called *pure Nash equilibria*. The global objective that is used in order to assess the efficiency of assignments is the *maximum completion time* over all jobs. A related quantity is the *makespan* (i.e., the maximum of the machine loads). Notice that when preemptive scheduling policies are used, these two quantities may not be the same (since idle time contributes to the completion time but not to the load of a machine). However, the optimal makespan is a lower bound on the optimal maximum completion time. The *price of anarchy* [42] is the maximum over all pure Nash equilibria of the ratio of the maximum completion time among all jobs over the optimal makespan. The *price of stability* [3] is the minimum over all pure Nash equilibria of the ratio of the maximum completion time among all jobs over the optimal makespan. The *approximation ratio* of a coordination mechanism is the maximum of the price of anarchy of the induced game over all input instances.

Four natural coordination mechanisms are the Makespan, Randomized, LongestFirst, and ShortestFirst. In the Makespan policy, each machine processes the jobs assigned to it “in parallel” so that the completion time of each job is the total load of the machine. Makespan is obviously a preemptive coordination mechanism. In the Randomized policy, the jobs are scheduled non-preemptively in random order. Here, the cost of each player is the expected completion time of her job. In the ShortestFirst and Longest-

First policies, the jobs assigned to a machine are scheduled in non-decreasing and non-increasing order of their processing times, respectively. In case of ties, a *global ordering* of the jobs according to their distinct IDs is used. This is necessary by any deterministic non-preemptive coordination mechanism in order to be well-defined. Note that no such information is required by the **Makespan** and **Randomized** policies; in this case, we say that they handle *anonymous jobs*. According to the terminology of [8], all these four coordination mechanisms are *strongly local* in the sense that the only information required by each machine in order to compute a schedule are the processing times of the jobs assigned to it. A *local* coordination mechanism may use all parameters (i.e., the load vector) of the jobs assigned to the same machine.

Designing coordination mechanisms with as small approximation ratio as possible is our main concern. But there are other issues related to efficiency. The price of anarchy is meaningful only in games where pure Nash equilibria exist. So, the primary goal of the designer of a coordination mechanism should be that the induced game *always has* pure Nash equilibria. Furthermore, these equilibria should be *easy to find*. A very interesting class of games in which the existence of pure Nash equilibria is guaranteed is that of *potential* games. These games have the property that a *potential function* can be defined on the states of the game so that in any two states differing in the strategy of a single player, the difference of the values of the potential function and the difference of the cost of the player have the same sign. This property guarantees that the state with minimum potential is a pure Nash equilibrium. Furthermore, it guarantees that, starting from any state, the system will reach (converge to) a pure Nash equilibrium after a finite number of *selfish moves*. Given a game, its *Nash dynamics* is a directed graph with the states of the game as nodes and edges connecting two states differing in the strategy of a single player if that player has an incentive to change her strategy according to the direction of the edge. The Nash dynamics of potential games do not contain any cycle. Another desirable property here is *fast* convergence, i.e., convergence to a pure Nash equilibrium in a polynomial number of selfish moves. A particular type of selfish moves that have been extensively considered in the literature [6, 15, 22, 39] is that of *best-response* moves. In a best-response move, a player having an incentive to change her strategy selects the strategy that yields the maximum decrease in her cost.

Potential games are strongly related to *congestion games* introduced by Rosenthal [43]. Rosenthal presented a potential function for these games with the following property: in any two states differing in the strategy of a single player, the difference of the values of the potential function *equals* the difference of the cost of the player. Monderer and Shapley [40] have proved that each potential game having this property is isomorphic to a congestion game. We point out that potential functions are not the only way to guarantee the existence of pure Nash equilibria. Several generalizations of congestion games such as those with player-specific latency functions [38] are not potential games but several subclasses of them provably have pure Nash equilibria.

Related work. The study of the price of anarchy of games began with the seminal work of Koutsoupias and Papadimitriou [35] and has played a central role in the recently emerging field of Algorithmic Game Theory [41]. Several papers provide bounds on the price of anarchy of different games of interest. Our work follows a different direction where the price of anarchy is the *objective to be minimized* and, in this sense, it is similar in spirit to studies where the main question is how to change the rules of the game at hand in order to improve the price of anarchy. Typical examples are the introduction of taxes or tolls in congestion games [11, 17, 24, 32, 51], protocol design in network and cost allocation games [12, 33], Stackelberg routing strategies [31, 34, 36, 44, 51], and network design [45].

Coordination mechanisms were introduced by Christodoulou, Koutsoupias, and Nanavati in [14]. They study the case where each player has the same load on each machine and, among other results, they consider the **LongestFirst** and **ShortestFirst** scheduling policies. We note that the **Makespan** and **Randomized** scheduling policies were used in [35] as models of selfish behaviour in scheduling, and since that paper,

the **Makespan** policy has been considered as standard in the study of selfish scheduling games in models simpler than the one of unrelated machines and is strongly related to the study of congestion games (see [52, 46] and the references therein). Immorlica et al. [29] study these four scheduling policies under several scheduling settings including the most general case of unrelated machines. They prove that the **Randomized** and **ShortestFirst** policies have approximation ratio $O(m)$ while the **LongestFirst** and **Makespan** policies have unbounded approximation ratio. Some scheduling policies are also related to earlier studies of local-search scheduling heuristics. So, the fact that the price of anarchy of the induced game may be unbounded follows by the work of Schuurman and Vredeveld [48]. As observed in [29], the equilibria of the game induced by **ShortestFirst** correspond to the solutions of the **ShortestFirst** scheduling heuristic which is known to be m -approximate [28]. The **Makespan** policy is known to induce potential games [20]. The **ShortestFirst** policy also induces potential games as proved in [29]. In Section 4, we present examples showing that the scheduling policies **LongestFirst** and **Randomized** do not induce potential games.¹

Azar et al. [8] study non-preemptive coordination mechanisms for unrelated machine scheduling. They prove that any local non-preemptive coordination mechanism is at least $\Omega(\log m)$ -approximate² while any strongly local non-preemptive coordination mechanism is at least $\Omega(m)$ -approximate; as a corollary, they solve an old open problem concerning the approximation ratio of the **ShortestFirst** heuristic. On the positive side, the authors of [8] present a non-preemptive local coordination mechanism (henceforth called **AJM-1**) that is $O(\log m)$ -approximate although it may induce games without pure Nash equilibria. The extra information used by this scheduling policy is the *inefficiency* of jobs (defined in the next section). They also present a technique that transforms this coordination mechanism to a preemptive one that induces potential games with price of anarchy $O(\log^2 m)$. In their mechanism, the players converge to a pure Nash equilibrium in n rounds of best-response moves. We will refer to this coordination mechanism as **AJM-2**. Both **AJM-1** and **AJM-2** use the IDs of the jobs.

Our results. We present three new coordination mechanisms for unrelated machine scheduling. Our mechanisms are deterministic, preemptive, and local. The schedules in each machine are computed as functions of the characteristics of jobs assigned to the machine, namely the load of jobs on the machine and their inefficiency. In all cases, the functions use an integer parameter $p \geq 1$; the best choice of this parameter for our coordination mechanisms is $p = O(\log m)$. Our analysis is heavily based on the convexity of simple polynomials and geometric inequalities for Euclidean norms.

Motivated by previous work, we first consider the scenario where jobs have distinct IDs. Our first coordination mechanism **ACOORD** uses this information and is superior to the known coordination mechanisms that induce games with pure Nash equilibria. The game induced is a potential game, has price of anarchy $\Theta(\log m)$, and the players converge to pure Nash equilibria in at most n rounds. Essentially, the equilibria of the game induced by **ACOORD** can be thought of as the solutions produced by the application of a particular online algorithm, similar to the greedy online algorithm for minimizing the ℓ_p norm of the machine loads [4, 10]. Interestingly, the local objective of the greedy online algorithm for the ℓ_p norm may not translate to a completion time of jobs in feasible schedules; the online algorithm implicit by **ACOORD** uses a different local objective that meets this constraint. The related results are presented in Section 3.

Next we address the case where no ID information is associated to the jobs (anonymous jobs). This scenario is relevant when the job owners do not wish to reveal their identities or in large-scale settings where distributing IDs to jobs is infeasible. Definitely, an advantage that could be used for coordination is lost in this way but this makes the problem of designing coordination mechanisms more challenging. In

¹After the appearance of the conference version of the paper, we became aware of two independent proofs that **Longest-First** may induce games that do not have pure Nash equilibria [19, 23].

²The corresponding proof of [8] contained an error which has been recently fixed by Fleischer and Svitkina [25].

Coordination mechanism	PoA	Pot.	PNE	IDs	Characteristics
ShortestFirst	$\Theta(m)$	Yes	Yes	Yes	Strongly local, non-preemptive
LongestFirst	unbounded	No	No	Yes	Strongly local, non-preemptive
Makespan	unbounded	Yes	Yes	No	Strongly local, preemptive
Randomized	$\Theta(m)$	No	?	No	Strongly local, non-preemptive
AJM-1	$\Theta(\log m)$	No	No	Yes	Local, non-preemptive
AJM-2	$O(\log^2 m)$	Yes	Yes	Yes	Local, preemptive, uses m
ACOORD	$\Theta(\log m)$	Yes	Yes	Yes	Local, preemptive, uses m
	$O(m^\epsilon)$	Yes	Yes	Yes	Local, preemptive
BCOORD	$O\left(\frac{\log m}{\log \log m}\right)$	No	?	No	Local, preemptive, uses m
	$O(m^\epsilon)$	No	?	No	Local, preemptive
CCOORD	$O(\log^2 m)$	Yes	Yes	No	Local, preemptive, uses m
	$O(m^\epsilon)$	Yes	Yes	No	Local, preemptive

Table 1: Comparison of our coordination mechanisms to previously known ones with respect to the price of anarchy of the induced game (PoA), whether they induced potential games or not (Pot.), the existence of pure Nash equilibria (PNE), and whether they use the job IDs or not.

Section 4, we present our second coordination mechanism **BCOORD** which induces a simple congestion game with player-specific polynomial latency functions of a particular form. The price of anarchy of this game is only $O\left(\frac{\log m}{\log \log m}\right)$. This result demonstrates that preemption may be useful in order to beat the $\Omega(\log m)$ lower bound of [8] for non-preemptive coordination mechanisms. On the negative side, we show that the game induced may not be a potential game by presenting an example where the Nash dynamics have a cycle.

Our third coordination mechanism **CCOORD** is presented in Section 5. The scheduling policy on each machine uses an interesting function on the loads of the jobs assigned to the machine and their inefficiency. The game induced by **CCOORD** is a potential game; the associated potential function is “cost-revealing” in the sense that it can be used to upper-bound the cost of equilibria. In particular, we show that the price of stability of the induced game is $O(\log m)$ and the price of anarchy is $O(\log^2 m)$. The coordination mechanism **CCOORD** is the first that handles anonymous jobs and simultaneously guarantees that the induced game is a potential game and has bounded price of anarchy. Table 1 compares our coordination mechanisms to the previously known ones.

Observe that the dependence of the parameter p on m requires that our mechanisms use the number of machines as input. By setting p equal to an appropriately large constant, our mechanisms achieve price of anarchy $O(m^\epsilon)$ for any constant $\epsilon > 0$. In particular, the coordination mechanisms **ACOORD** and **CCOORD** are the first ones that do not use the number of machines as a parameter, induce games with pure Nash equilibria, and have price of anarchy $o(m)$.

We remark that the current paper contains several improvements compared to its conference version. There, the three coordination mechanisms had the restriction that a job with inefficiency more than m on some machine has infinite completion time when assigned to that machine. Here, we have removed this restriction and have adapted the analysis accordingly. A nice consequence of the new definition is that the coordination mechanisms can now be defined so that they do not use the number of machines as a parameter. Furthermore, the definition of **ACOORD** has been significantly simplified. Also, the analysis of the price of anarchy of the coordination mechanism **BCOORD** in the conference version used a technical lemma which is implicit in [50]. In the current version, we present a different self-contained proof that is based on convexity properties of polynomials and Minkowski inequality; the new proof has a similar structure with

the analysis of the price of anarchy of mechanism ACOORD.

We begin with preliminary technical definitions in Section 2 and conclude with interesting open questions in Section 6.

2 Preliminaries

In this section, we present our notation and give some statements that will be useful later. We reserve n and m for the number of jobs and machines, respectively, and the indices i and j for jobs and machines, respectively. Unless specified otherwise, the sums \sum_i and \sum_j run over all jobs and over all machines, respectively. Assignments are denoted by N or O . With some abuse in notation, we use N_j to denote both the set of jobs assigned to machine j and the set of their loads on machine j . We use the notation $L(N_j)$ to denote the load of machine j under the assignment N . More generally, $L(A)$ denotes the sum of the elements for any set of non-negative reals A . For an assignment N which assigns job i to machine j , we denote the completion time of job i under a given scheduling policy by $\mathcal{P}(i, N_j)$. Note that, besides defining the completion times, we do not discuss the particular way the jobs are scheduled by the scheduling policies we present. However, we require that *feasible* schedules are computable efficiently. A natural sufficient and necessary condition is the following: for any job $i \in N_j$, the total load of jobs with completion time at most $\mathcal{P}(i, N_j)$ is at most $\mathcal{P}(i, N_j)$.

Our three coordination mechanisms use the inefficiency of jobs in order to compute schedules. We denote by $w_{i,\min}$ the minimum load of job i over all machines. Then, its inefficiency ρ_{ij} on machine j is defined as $\rho_{ij} = w_{ij}/w_{i,\min}$.

Our proofs are heavily based on the convexity of simple polynomials such as z^k for $k \geq 1$ and on the relation of Euclidean norms of the machine loads and the makespan. Recall that the ℓ_k norm of the machine loads for an assignment N is $(\sum_j L(N_j)^k)^{1/k}$. The proof of the next lemma is trivial.

Lemma 1 *For any assignment N , $\max_j L(N_j) \leq (\sum_j L(N_j)^k)^{1/k} \leq m^{1/k} \max_j L(N_j)$.*

In some of the proofs, we also use the Minkowski inequality (or the triangle inequality for the ℓ_p norm).

Lemma 2 (Minkowski inequality) *$(\sum_{t=1}^s (a_t + b_t)^k)^{1/k} \leq (\sum_{t=1}^s a_t^k)^{1/k} + (\sum_{t=1}^s b_t^k)^{1/k}$, for any $k \geq 1$ and $a_t, b_t \geq 0$.*

The following two technical lemmas are used in some of our proofs. We include them here for easy reference.

Lemma 3 *Let $r \geq 1$, $t \geq 0$ and $a_i \geq 0$, for $i = 1, \dots, k$. Then,*

$$\sum_{i=1}^k ((t + a_i)^r - t^r) \leq \left(t + \sum_{i=1}^k a_i \right)^r - t^r$$

Proof. The case when $a_i = 0$ for $i = 1, \dots, k$ is trivial. Assume otherwise and let $\xi = \sum_{i=1}^k a_i$ and $\xi_i = a_i/\xi$. Clearly, $\sum_{i=1}^k \xi_i = 1$. By the convexity of function z^r in $[0, \infty)$, we have that

$$\begin{aligned} (t + a_i)^r &= \left((1 - \xi_i)t + \xi_i \left(t + \sum_{i=1}^k a_i \right) \right)^r \\ &\leq (1 - \xi_i)t^r + \xi_i \left(t + \sum_{i=1}^k a_i \right)^r \end{aligned} \tag{1}$$

for $i = 1, \dots, k$. Using (1), we obtain

$$\begin{aligned} \sum_{i=1}^k ((t + a_i)^r - t^r) &\leq t^r \left(\sum_{i=1}^k (1 - \xi_i) - k \right) + \left(t + \sum_{i=1}^k a_i \right)^r \sum_{i=1}^k \xi_i \\ &= \left(t + \sum_{i=1}^k a_i \right)^r - t^r \end{aligned}$$

■

Lemma 4 For any $z_0 \geq 0$, $\alpha \geq 0$, and $p \geq 1$, it holds

$$(p+1)\alpha z_0^p \leq (z_0 + \alpha)^{p+1} - z_0^{p+1} \leq (p+1)\alpha(z_0 + \alpha)^p.$$

Proof. The inequality trivially holds if $\alpha = 0$. If $\alpha > 0$, the inequality follows since, due to the convexity of the function z^{p+1} , the slope of the line that crosses points (z_0, z_0^{p+1}) and $(z_0 + \alpha, (z_0 + \alpha)^{p+1})$ is between its derivative at points z_0 and $z_0 + \alpha$. ■

We also refer to the multinomial and binomial theorems. [27] provides an extensive overview of the inequalities we use and their history (see also wikipedia.org for a quick survey).

3 The coordination mechanism ACOORD

The coordination mechanism ACOORD uses a global ordering of the jobs according to their distinct IDs. Without loss of generality, we may assume that the index of a job is its ID. Let N be an assignment and denote by N^i the restriction of N to the jobs with the i smallest IDs. ACOORD schedules job i on machine j so that it completes at time

$$\mathcal{P}(i, N_j) = (\rho_{ij})^{1/p} L(N_j^i).$$

Since $\rho_{ij} \geq 1$, the schedules produced are always feasible.

Consider the sequence of jobs in increasing order of their IDs and assume that each job plays a best-response move. In this case, job i will select that machine j so that the quantity $(\rho_{ij})^{1/p} L(N_j^i)$ is minimized. Since the completion time of job i depends only on jobs with smaller IDs, no job will have an incentive to change its strategy and the resulting assignment is a pure Nash equilibrium. The following lemma extends this observation in a straightforward way.

Lemma 5 The game induced by the coordination mechanism ACOORD is a potential game. Furthermore, any sequence of n rounds of best-response moves converges to a pure Nash equilibrium.

Proof. Notice that since a job does not affect the completion time of jobs with smaller IDs, the vector of completion times of the jobs (sorted in increasing order of their IDs) decreases lexicographically when a job improves its cost by deviating to another strategy and, hence, it is a potential function for the game induced by the coordination mechanism ACOORD.

Now, consider n rounds of best-response moves of the jobs in the induced game such that each job plays at least once in each round. It is not hard to see that after round i , the job i will have selected that machine j so that the quantity $(\rho_{ij})^{1/p} L(N_j^i)$ is minimized. Since the completion time of job i depends only on jobs with smaller IDs, job i has no incentive to move after round i and, hence, no job will have an incentive to change its strategy after the n rounds. So, the resulting assignment is a pure Nash equilibrium. ■

The sequence of best-response moves mentioned above can be thought of as an online algorithm that processes the jobs in increasing order of their IDs. The local objective is slightly different than the local objective of the greedy online algorithm for minimizing the ℓ_{p+1} norm of the machine loads [7, 10]; in that algorithm, job i is assigned to a machine j so that the quantity $(L(N_j^{i-1}) + w_{ij})^{p+1} - L(N_j^{i-1})^{p+1}$ is minimized. Here, we remark that we do not see how the local objective of that algorithm could be simulated by a scheduling policy that always produces feasible schedules. This constraint is trivially satisfied by the coordination mechanism **ACOORD**. The next lemma bounds the maximum completion time at pure Nash equilibria in terms of the ℓ_{p+1} norm of the machine loads and the optimal makespan.

Lemma 6 *Let N be a pure Nash equilibrium of the game induced by the coordination mechanisms **ACOORD** and let O be an optimal assignment. Then*

$$\max_{j,i \in N_j} \mathcal{P}(i, N_j) \leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j).$$

Proof. Let i^* be the job that has the maximum completion time in assignment N . Denote by j_1 the machine i^* uses in N and let j_2 be a machine such that $\rho_{i^*j_2} = 1$. If $j_1 = j_2$, the definition of the coordination mechanism **ACOORD** yields

$$\begin{aligned} \max_{j,i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\ &= L(N_{j_1}^{i^*}) \\ &\leq L(N_{j_1}) \\ &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}}. \end{aligned}$$

Otherwise, since player i^* has no incentive to use machine j_2 instead of j_1 , we have

$$\begin{aligned} \max_{j,i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\ &\leq \mathcal{P}(i^*, N_{j_2} \cup \{w_{i^*j_2}\}) \\ &= L(N_{j_2}^{i^*}) + w_{i^*j_2} \\ &\leq L(N_{j_2}) + \min_j w_{i^*j} \\ &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j). \end{aligned}$$

■

Next we show that the approximation ratio of **ACOORD** is $O(\log m)$ (for well-selected values of the parameter p). The analysis borrows and extends techniques from the analysis of the greedy online algorithm for the ℓ_p norm in [10].

Theorem 7 *The price of anarchy of the game induced by the coordination mechanism **ACOORD** with $p = \Theta(\log m)$ is $O(\log m)$. Also, for every constant $\epsilon \in (0, 1/2]$, the price of anarchy of the game induced by the coordination mechanism **ACOORD** with $p = 1/\epsilon - 1$ is $O(m^\epsilon)$.*

Proof. Consider a pure Nash equilibrium N and an optimal assignment O . Since no job has an incentive to change her strategy from N , for any job i that is assigned to machine j_1 in N and to machine j_2 in O , by the definition of ACOORD we have that

$$(\rho_{ij_1})^{1/p} L(N_{j_1}^i) \leq (\rho_{ij_2})^{1/p} (L(N_{j_2}^{i-1}) + w_{ij_2}).$$

Equivalently, by raising both sides to the power p and multiplying with $w_{i,\min}$, we have that

$$w_{ij_1} L(N_{j_1}^i)^p \leq w_{ij_2} (L(N_{j_2}^{i-1}) + w_{ij_2})^p.$$

Using the binary variables x_{ij} and y_{ij} to denote whether job i is assigned to machine j in the assignment N ($x_{ij} = 1$) and O ($y_{ij} = 1$), respectively, or not ($x_{ij} = 0$ and $y_{ij} = 0$, respectively), we can express this last inequality as follows.

$$\sum_j x_{ij} w_{ij} L(N_j^i)^p \leq \sum_j y_{ij} w_{ij} (L(N_j^{i-1}) + w_{ij})^p$$

By summing over all jobs and multiplying with $(e-1)(p+1)$, we have

$$\begin{aligned} & (e-1)(p+1) \sum_i \sum_j x_{ij} w_{ij} L(N_j^i)^p \\ & \leq (e-1)(p+1) \sum_i \sum_j y_{ij} w_{ij} (L(N_j^{i-1}) + w_{ij})^p \\ & \leq (e-1)(p+1) \sum_j \sum_i y_{ij} w_{ij} (L(N_j) + w_{ij})^p \\ & = (e-1)(p+1) \sum_j \sum_i y_{ij} w_{ij} (L(N_j) + y_{ij} w_{ij})^p \\ & \leq \sum_j \sum_i \left((L(N_j) + e y_{ij} w_{ij})^{p+1} - (L(N_j) + y_{ij} w_{ij})^{p+1} \right) \\ & \leq \sum_j \sum_i \left((L(N_j) + e y_{ij} w_{ij})^{p+1} - L(N_j)^{p+1} \right) \\ & \leq \sum_j \left(\left(L(N_j) + e \sum_i y_{ij} w_{ij} \right)^{p+1} - L(N_j)^{p+1} \right) \\ & = \sum_j (L(N_j) + e L(O_j))^{p+1} - \sum_j L(N_j)^{p+1} \\ & \leq \left(\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + e \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \right)^{p+1} - \sum_j L(N_j)^{p+1}. \end{aligned} \quad (2)$$

The second inequality follows by exchanging the sums and since $L(N_j^{i-1}) \leq L(N_j)$, the first equality follows since $y_{ij} \in \{0, 1\}$, the third inequality follows by applying Lemma 4 with $\alpha = (e-1)y_{ij}w_{ij}$ and $z_0 = L(N_j) + y_{ij}w_{ij}$, the fourth inequality is obvious, the fifth inequality follows by Lemma 3, the second equality follows since the definition of the variables y_{ij} implies that $L(O_j) = \sum_i y_{ij}w_{ij}$, and the last inequality follows by Minkowski inequality (Lemma 2).

Now, we will relate the ℓ_{p+1} norm of the machines loads of assignments N and O . We have

$$\begin{aligned}
(e-1) \sum_j L(N_j)^{p+1} &= (e-1) \sum_j L(N_j^n)^{p+1} \\
&= (e-1) \sum_{i=1}^n \sum_j \left(L(N_j^i)^{p+1} - L(N_j^{i-1})^{p+1} \right) \\
&= (e-1) \sum_{i=1}^n \sum_j \left(L(N_j^i)^{p+1} - (L(N_j^i) - x_{ij}w_{ij})^{p+1} \right) \\
&\leq (e-1)(p+1) \sum_i \sum_j x_{ij}w_{ij} L(N_j^i)^p \\
&\leq \left(\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + e \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \right)^{p+1} - \sum_j L(N_j)^{p+1}.
\end{aligned}$$

The first two equalities are obvious (observe that $L(N_j^0) = 0$), the third one follows by the definition of variables x_{ij} , the first inequality follows by applying Lemma 4 with $\alpha = x_{ij}w_{ij}$ and $z_0 = L(N_j^i) - x_{ij}w_{ij}$, and the last inequality follows by inequality (2).

So, the above inequality yields

$$\begin{aligned}
\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} &\leq \frac{e}{e^{\frac{1}{p+1}} - 1} \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \\
&\leq e(p+1) \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \\
&\leq e(p+1)m^{\frac{1}{p+1}} \max_j L(O_j).
\end{aligned}$$

The second inequality follows since $e^z \geq z + 1$ for $z \geq 0$ and the third one follows by Lemma 1.

Now, using Lemma 6 and this last inequality, we obtain that

$$\begin{aligned}
\max_{j, i \in N_j} \mathcal{P}(i, N_j) &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j) \\
&\leq \left(e(p+1)m^{\frac{1}{p+1}} + 1 \right) \max_j L(O_j).
\end{aligned}$$

The desired bounds follow by setting $p = \Theta(\log m)$ and $p = 1/\epsilon - 1$, respectively. ■

Our logarithmic bound is asymptotically tight; this follows by the connection to online algorithms mentioned above and the lower bound of [9].

4 The coordination mechanism BCOORD

We now turn our attention to coordination mechanisms that handle anonymous jobs. We define the coordination mechanism BCOORD by slightly changing the definition of ACOORD so that the completion time

of a job does not depend on its ID. So, **BCOORD** schedules job i on machine j so that it finishes at time

$$\mathcal{P}(i, N_j) = (\rho_{ij})^{1/p} L(N_j).$$

Since $\rho_{ij} \geq 1$, the schedules produced are always feasible. The next lemma bounds the maximum completion time at pure Nash equilibria (again in terms of the ℓ_{p+1} norm of the machine loads and the optimal makespan).

Lemma 8 *Let N be a pure Nash equilibrium of the game induced by the coordination mechanisms **BCOORD** and let O be an optimal assignment. Then*

$$\max_{j, i \in N_j} \mathcal{P}(i, N_j) \leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j).$$

Proof. The proof is almost identical to the proof of Lemma 6; we include it here for completeness. Let i^* be the job that has the maximum completion time in assignment N . Denote by j_1 the machine i^* uses in N and let j_2 be a machine such that $\rho_{i^*j_2} = 1$. If $j_1 = j_2$, the definition of the coordination mechanism **BCOORD** yields

$$\begin{aligned} \max_{j, i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\ &= L(N_{j_1}) \\ &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}}. \end{aligned}$$

Otherwise, since player i^* has no incentive to use machine j_2 instead of j_1 , we have

$$\begin{aligned} \max_{j, i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\ &\leq \mathcal{P}(i^*, N_{j_2} \cup \{w_{i^*j_2}\}) \\ &= L(N_{j_2}) + w_{i^*j_2} \\ &= L(N_{j_2}) + \min_j w_{i^*j} \\ &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j). \end{aligned}$$

■

We are ready to present our upper bounds on the price of anarchy of the induced game.

Theorem 9 *The price of anarchy of the game induced by the coordination mechanism **BCOORD** with $p = \Theta(\log m)$ is $O\left(\frac{\log m}{\log \log m}\right)$. Also, for every constant $\epsilon \in (0, 1/2]$, the price of anarchy of the game induced by the coordination mechanism **BCOORD** with $p = 1/\epsilon - 1$ is $O(m^\epsilon)$.*

Proof. Consider a pure Nash equilibrium N and an optimal assignment O . Since no job has an incentive to change her strategy from N , for any job i that is assigned to machine j_1 in N and to machine j_2 in O , we have that

$$(\rho_{ij_1})^{1/p} L(N_{j_1}) \leq (\rho_{ij_2})^{1/p} (L(N_{j_2}) + w_{ij_2}).$$

Equivalently, by raising both sides to the power p and multiplying both sides with $w_{i,\min}$, we have that

$$w_{ij_1} L(N_{j_1})^p \leq w_{ij_2} (L(N_{j_2}) + w_{ij_2})^p.$$

Using the binary variables x_{ij} and y_{ij} to denote whether job i is assigned to machine j in the assignments N ($x_{ij} = 1$) and O ($y_{ij} = 1$), respectively, or not ($x_{ij} = 0$ and $y_{ij} = 0$, respectively), we can express this last inequality as follows:

$$\sum_j x_{ij} w_{ij} L(N_j)^p \leq \sum_j y_{ij} w_{ij} (L(N_j) + w_{ij})^p.$$

By summing over all jobs and multiplying with p , we have

$$\begin{aligned} & p \sum_i \sum_j x_{ij} w_{ij} L(N_j)^p \\ & \leq p \sum_i \sum_j y_{ij} w_{ij} (L(N_j) + w_{ij})^p \\ & = p \sum_j \sum_i y_{ij} w_{ij} (L(N_j) + y_{ij} w_{ij})^p \\ & \leq \sum_j \sum_i \left(\left(L(N_j) + \frac{2p+1}{p+1} y_{ij} w_{ij} \right)^{p+1} - (L(N_j) + y_{ij} w_{ij})^{p+1} \right) \\ & \leq \sum_j \sum_i \left(\left(L(N_j) + \frac{2p+1}{p+1} y_{ij} w_{ij} \right)^{p+1} - L(N_j)^{p+1} \right) \\ & \leq \sum_j \left(\left(L(N_j) + \frac{2p+1}{p+1} \sum_i y_{ij} w_{ij} \right)^{p+1} - L(N_j)^{p+1} \right) \\ & = \sum_j \left(\left(L(N_j) + \frac{2p+1}{p+1} L(O_j) \right)^{p+1} - L(N_j)^{p+1} \right) \\ & \leq \left(\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \frac{2p+1}{p+1} \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \right)^{p+1} - \sum_j L(N_j)^{p+1} \end{aligned} \quad (3)$$

The first equality follows by exchanging the sums and since $y_{ij} \in \{0, 1\}$, the second inequality follows by applying Lemma 4 with $\alpha = \frac{p}{p+1} y_{ij} w_{ij}$ and $z_0 = L(N_j) + y_{ij} w_{ij}$, the third inequality is obvious, the fourth inequality follows by applying Lemma 3, the second equality follows since the definition of variables y_{ij} implies that $L(O_j) = \sum_i y_{ij} w_{ij}$, and the last inequality follows by applying Minkowski inequality (Lemma 2).

Now, we relate the ℓ_{p+1} norm of the machine loads of assignments N and O . We have

$$(p+1) \sum_j L(N_j)^{p+1} = p \sum_j L(N_j)^{p+1} + \sum_j L(N_j)^{p+1}$$

$$\begin{aligned}
&= p \sum_i \sum_j x_{ij} w_{ij} L(N_j)^p + \sum_j L(N_j)^{p+1} \\
&\leq \left(\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \frac{2p+1}{p+1} \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \right)^{p+1}.
\end{aligned}$$

The first equality is obvious, the second one follows by the definition of variables x_{ij} and the inequality follows by inequality (3).

So, the above inequalities yield

$$\begin{aligned}
\left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} &\leq \frac{2p+1}{p+1} \frac{1}{(p+1)^{\frac{1}{p+1}} - 1} \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \\
&\leq \frac{2p+1}{\ln(p+1)} \left(\sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} \\
&\leq \frac{2p+1}{\ln(p+1)} m^{\frac{1}{p+1}} \max_j L(O_j).
\end{aligned}$$

The second inequality follows since $e^z \geq z + 1$ for $z \geq 0$ and the third one follows by Lemma 1.

Now, using Lemma 8 and our last inequality we have

$$\begin{aligned}
\max_{j, i \in N_j} \mathcal{P}(i, N_j) &\leq \left(\sum_j L(N_j)^{p+1} \right)^{\frac{1}{p+1}} + \max_j L(O_j) \\
&\leq \left(1 + \frac{2p+1}{\ln(p+1)} m^{\frac{1}{p+1}} \right) \max_j L(O_j).
\end{aligned}$$

The desired bounds follows by setting $p = \Theta(\log m)$ and $p = 1/\epsilon - 1$, respectively. ■

Note that the game induced by **BCOORD** with $p = 1$ is the same with the game induced by the coordination mechanism **CCOORD** (with $p = 1$) that we present in the next section. As such, it also has a potential function (also similar to the potential function of [26] for linear weighted congestion games) as we will see in Lemma 13. In this way, we obtain a coordination mechanism that induces a potential game, handles anonymous jobs, and has approximation ratio $O(\sqrt{m})$. Unfortunately, the next theorem demonstrates that, for higher values of p , the Nash dynamics of the game induced by **BCOORD** may contain a cycle.

Theorem 10 *The game induced by the coordination mechanism **BCOORD** with $p = 2$ is not a potential game.*

Before proving Theorem 10, we show that the games induced by the coordination mechanisms **LongestFirst** and **Randomized** may not be potential games either. All the instances presented in the following consist of four machines and three basic jobs A , B , and C . In each case, we show that the Nash dynamics contain a cycle of moves of the three basic jobs.

First consider the **LongestFirst** policy and the instance depicted in the following table.

	A	B	C
1	14	∞	5
2	∞	10	∞
3	3	9	10
4	7	8	9

The cycle is defined on the following states:

$$(C, \underline{B}, A,) \rightarrow (C, , \underline{A}B,) \rightarrow (C, , \underline{B}, A) \rightarrow (C, , , \underline{A}B) \rightarrow (A\underline{C}, , , B) \rightarrow$$

$$(\underline{A}, , C, B) \rightarrow (, , A\underline{C}, B) \rightarrow (, , A, \underline{B}C) \rightarrow (, B, A, \underline{C}) \rightarrow (C, B, A,).$$

Notice that the first and last assignment are the same. In each state, the player that moves next is underlined. Job B is at machine 2 in the first assignment and has completion time 10. Hence, it has an incentive to move to machine 3 (second assignment) where its completion time is 9. Job A has completion time 12 in the second assignment since it is scheduled after job B which has higher load on machine 3. Moving to machine 4 (third assignment), it decreases its completion time to 7. The remaining moves in the cycle can be verified accordingly.

The instance for the **Randomized** policy contains four additional jobs D , E , F , and G which are always scheduled on machines 1, 2, 3, and 4, respectively (i.e., they have infinite load on the other machines). It is depicted in the following table.

	A	B	C	D	E	F	G
1	80	∞	100	2	∞	∞	∞
2	∞	171	∞	∞	2	∞	∞
3	2	154	124	∞	∞	32	∞
4	2	76	10	∞	∞	∞	184

The cycle is defined by the same moves of the basic jobs as in the case of **LongestFirst**:

$$(CD, \underline{B}E, AF, G) \rightarrow (CD, E, \underline{A}BF, G) \rightarrow (CD, E, \underline{B}F, AG) \rightarrow (CD, E, F, \underline{A}BG) \rightarrow$$

$$(A\underline{C}D, E, F, BG) \rightarrow (\underline{A}D, E, CF, BG) \rightarrow (D, E, A\underline{C}F, BG) \rightarrow (D, E, AF, \underline{B}CG) \rightarrow$$

$$(D, BE, AF, \underline{C}G) \rightarrow (CD, BE, AF, G).$$

Recall that (see [29, 35]) the expected completion time of a job i which is scheduled on machine j in an assignment N is $\frac{1}{2}(w_{ij} + L(N_j))$ when the **Randomized** policy is used. In each state, the player that moves next is underlined. It can be easily verified that each player in this cycle improves her cost by exactly 1. For example, job B has expected completion time $\frac{1}{2}(171 + 171 + 2) = 172$ at machine 2 in the first assignment and, hence, an incentive to move to machine 3 in the second assignment where its completion time is $\frac{1}{2}(154 + 2 + 154 + 32) = 171$.

Proof of Theorem 10. Besides the three basic jobs, the instance for the **BCOORD** policy with $p = 2$ contains two additional jobs D and E which are always scheduled on machines 3 and 4, respectively. The instance is depicted in the following table.

	A	B	C	D	E
1	4.0202	∞	4.0741	∞	∞
2	∞	8.2481	∞	∞	∞
3	0.0745	0.6302	0.3078	29.1331	∞
4	2.4447	5.1781	2.4734	∞	2.7592

The cycle is defined by the same moves of the basic jobs as in the previous cases:

$$(C, \underline{B}, AD, E) \rightarrow (C, , \underline{ABD}, E) \rightarrow (C, , \underline{BD}, AE) \rightarrow (C, , D, \underline{ABE}) \rightarrow (AC, , D, BE) \rightarrow$$

$$(\underline{A}, , CD, BE) \rightarrow (, , AC, D, BE) \rightarrow (, , AD, \underline{BCE}) \rightarrow (, B, AD, \underline{CE}) \rightarrow (C, B, AD, E).$$

Notice that, instead of considering the completion time $(\rho_{ij})^{1/p} L(N_j)$ of a job i on machine j in an assignment N , it is equivalent to consider its cost as $w_{ij} L(N_j)^p$. In this way, we can verify that in any of the moves in the above cycle, the job that moves improves its cost. For example, job B has cost $8.2481^3 = 561.127758090641$ on machine 2 in the first assignment and cost $0.6302(0.0745 + 0.6302 + 29.1331)^2 = 561.063473430968$ on machine 3 in the second assignment. ■

5 The coordination mechanism CCOORD

In this section we present and analyze the coordination mechanism CCOORD that handles anonymous jobs and guarantees that the induced game has pure Nash equilibria, price of anarchy at most $O(\log^2 m)$, and price of stability $O(\log m)$. In order to define the scheduling policy, we first define an interesting family of functions.

Definition 11 For integer $k \geq 0$, the function Ψ_k mapping finite sets of reals to the reals is defined as follows: $\Psi_k(\emptyset) = 0$ for any integer $k \geq 1$, $\Psi_0(A) = 1$ for any (possibly empty) set A , and for any non-empty set $A = \{a_1, a_2, \dots, a_n\}$ and integer $k \geq 1$,

$$\Psi_k(A) = k! \sum_{1 \leq d_1 \leq \dots \leq d_k \leq n} \prod_{t=1}^k a_{d_t}.$$

So, $\Psi_k(A)$ is essentially the sum of all possible monomials of total degree k on the elements of A . Each term in the sum has coefficient $k!$. Clearly, $\Psi_1(A) = L(A)$. For $k \geq 2$, compare $\Psi_k(A)$ with $L(A)^k$ which can also be expressed as the sum of the same terms, albeit with different coefficients in $\{1, \dots, k!\}$, given by the multinomial theorem.

The coordination mechanism CCOORD schedules job i on machine j in an assignment N so that its completion time is

$$\mathcal{P}(i, N_j) = (\rho_{ij} \Psi_p(N_j))^{1/p}.$$

Our proofs extensively use the properties in the next lemma; its proof is given in appendix. The first inequality implies that the schedule defined by CCOORD is always feasible.

Lemma 12 For any integer $k \geq 1$, any finite set of non-negative reals A , and any non-negative real b the following hold:

$$\begin{array}{ll} a. L(A)^k \leq \Psi_k(A) \leq k! L(A)^k & d. \Psi_k(A \cup \{b\}) - \Psi_k(A) = kb \Psi_{k-1}(A \cup \{b\}) \\ b. \Psi_{k-1}(A)^k \leq \Psi_k(A)^{k-1} & e. \Psi_k(A) \leq k L(A) \Psi_{k-1}(A) \\ c. \Psi_k(A \cup \{b\}) = \sum_{t=0}^k \frac{k!}{(k-t)!} b^t \Psi_{k-t}(A) & f. \Psi_k(A \cup \{b\}) \leq \left(\Psi_k(A)^{1/k} + \Psi_k(\{b\})^{1/k} \right)^k \end{array}$$

The second property implies that $\Psi_k(A)^{1/k} \leq \Psi_{k'}(A)^{1/k'}$ for any integer $k' \geq k$. The third property suggests an algorithm for computing $\Psi_k(A)$ in time polynomial in k and $|A|$ using dynamic programming.

A careful examination of the definitions of the coordination mechanisms **BCOORD** and **CCOORD** and property (a) in the above lemma, reveals that **CCOORD** makes the completion time of a job assigned to machine j dependent on the approximation $\Psi_p(N_j)^{1/p}$ of the load $L(N_j)$ of the machine instead of its exact load as **BCOORD** does. This will be the crucial tool in order to guarantee that the induced game is a potential game without significantly increasing the price of anarchy. The next lemma defines a potential function on the states of the induced game that will be very useful later.

Lemma 13 *The function $\Phi(N) = \sum_j \Psi_{p+1}(N_j)$ is a potential function for the game induced by the coordination mechanism **CCOORD**. Hence, this game always has a pure Nash equilibrium.*

Proof. Consider two assignments N and N' differing in the strategy of the player controlling job i . Assume that job i is assigned to machine j_1 in N and to machine $j_2 \neq j_1$ in N' . Observe that $N_{j_1} = N'_{j_1} \cup \{w_{ij_1}\}$ and $N'_{j_2} = N_{j_2} \cup \{w_{ij_2}\}$. By Lemma 12d, we have that $\Psi_{p+1}(N_{j_1}) - \Psi_{p+1}(N'_{j_1}) = (p+1)w_{ij_1}\Psi_p(N_{j_1})$ and $\Psi_{p+1}(N'_{j_2}) - \Psi_{p+1}(N_{j_2}) = (p+1)w_{ij_2}\Psi_p(N'_{j_2})$. Using these properties and the definitions of the coordination mechanism **CCOORD** and function Φ , we have

$$\begin{aligned} \Phi(N) - \Phi(N') &= \sum_j \Psi_{p+1}(N_j) - \sum_j \Psi_{p+1}(N'_j) \\ &= \Psi_{p+1}(N_{j_1}) + \Psi_{p+1}(N_{j_2}) - \Psi_{p+1}(N'_{j_1}) - \Psi_{p+1}(N'_{j_2}) \\ &= (p+1)w_{ij_1}\Psi_p(N_{j_1}) - (p+1)w_{ij_2}\Psi_p(N'_{j_2}) \\ &= (p+1)w_{i,\min} \left(\mathcal{P}(i, N_{j_1})^p - \mathcal{P}(i, N'_{j_2})^p \right) \end{aligned}$$

which means that the difference of the potentials of the two assignments and the difference of the completion time of player i have the same sign as desired. \blacksquare

The next lemma relates the maximum completion time of a pure Nash equilibrium to the optimal makespan provided that their potentials are close.

Lemma 14 *Let O be an optimal assignment and let N be a pure Nash equilibrium of the game induced by the coordination mechanism **CCOORD** such that $(\Phi(N))^{\frac{1}{p+1}} \leq \gamma (\Phi(O))^{\frac{1}{p+1}}$. Then,*

$$\max_{j, i \in N_j} \mathcal{P}(i, N_j) \leq \left(\gamma(p+1)m^{\frac{1}{p+1}} + p \right) \max_j L(O_j).$$

Proof. Let i^* be the job that has the maximum completion time in N . Denote by j_1 the machine i^* uses in assignments N and let j_2 be a machine such that $\rho_{i^*j_2} = 1$. If $j_1 = j_2$, the definition of the coordination mechanism **CCOORD** and Lemma 12b yield

$$\begin{aligned} \max_{j, i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\ &= \Psi_p(N_{j_1})^{1/p} \\ &\leq \Psi_{p+1}(N_{j_1})^{\frac{1}{p+1}} \\ &\leq \left(\sum_j \Psi_{p+1}(N_j) \right)^{\frac{1}{p+1}}. \end{aligned} \tag{4}$$

Otherwise, since player i has no incentive to use machine j_2 instead of j_1 , we have

$$\begin{aligned}
\max_{j, i \in N_j} \mathcal{P}(i, N_j) &= \mathcal{P}(i^*, N_{j_1}) \\
&\leq \mathcal{P}(i^*, N_{j_2} \cup \{w_{i^* j_2}\}) \\
&= \Psi_p(N_{j_2} \cup \{w_{i^* j_2}\})^{1/p} \\
&\leq \Psi_p(N_{j_2})^{1/p} + \Psi_p(\{w_{i^* j_2}\})^{1/p} \\
&\leq \Psi_{p+1}(N_{j_2})^{\frac{1}{p+1}} + (p!)^{1/p} w_{i^* j_2} \\
&= \Psi_{p+1}(N_{j_2})^{\frac{1}{p+1}} + (p!)^{1/p} \min_j w_{i^* j} \\
&\leq \left(\sum_j \Psi_{p+1}(N_j) \right)^{\frac{1}{p+1}} + p \max_j L(O_j). \tag{5}
\end{aligned}$$

The first two equalities follows by the definition of **CCOORD**, the first inequality follows since player i^* has no incentive to use machine j_2 instead of j_1 , the second inequality follows by Lemma 12f, the third inequality follows by Lemma 12b and the definition of function Ψ_p , the third equality follows by the definition of machine j_2 and the last inequality is obvious.

Now, observe that the term in parenthesis in the rightmost side of inequalities (4) and (5) equals the potential $\Phi(N)$. Hence, in any case, we have

$$\begin{aligned}
\max_{j, i \in N_j} \mathcal{P}(i, N_j) &\leq (\Phi(N))^{\frac{1}{p+1}} + p \max_j L(O_j) \\
&\leq \gamma(\Phi(O))^{\frac{1}{p+1}} + p \max_j L(O_j) \\
&= \gamma \left(\sum_j \Psi_{p+1}(O_j) \right)^{\frac{1}{p+1}} + p \max_j L(O_j) \\
&\leq \gamma \left((p+1)! \sum_j L(O_j)^{p+1} \right)^{\frac{1}{p+1}} + p \max_j L(O_j) \\
&\leq \left(\gamma(p+1)m^{\frac{1}{p+1}} + p \right) \max_j L(O_j).
\end{aligned}$$

The second inequality follows by the inequality on the potentials of assignments N and O , the equality follows by the definition of the potential function Φ , the third inequality follows by Lemma 12a and the last one follows by Lemma 1. \blacksquare

A first application of Lemma 14 is in bounding the price of stability of the induced game.

Theorem 15 *The game induced by the coordination mechanism **CCOORD** with $p = \Theta(\log m)$ has price of stability at most $O(\log m)$.*

Proof. Consider the optimal assignment O and the pure Nash equilibrium N of minimum potential. We have $(\Phi(N))^{\frac{1}{p+1}} \leq (\Phi(O))^{\frac{1}{p+1}}$ and, using Lemma 14, we obtain that the maximum completion time in N is at most $(p+1)m^{\frac{1}{p+1}} + p$ times the makespan of O . Setting $p = \Theta(\log m)$, the theorem follows. \blacksquare

A second application of Lemma 14 is in bounding the price of anarchy. In order to apply it, we need a relation between the potential of an equilibrium and the potential of an optimal assignment; this is provided by the next lemma.

Lemma 16 *Let O be an optimal assignment and N be a pure Nash equilibrium of the game induced by the coordination mechanism CCOORD. Then,*

$$(\Phi(N))^{\frac{1}{p+1}} \leq \frac{p+1}{\ln 2} (\Phi(O))^{\frac{1}{p+1}}.$$

Proof. Consider a pure Nash equilibrium N and an optimal assignment O . Since no job has an incentive to change her strategy from N , for any job i that is assigned to machine j_1 in N and to machine j_2 in O , we have that

$$(\rho_{ij_1} \Psi_p(N_{j_1}))^{1/p} \leq (\rho_{ij_2} \Psi_p(N_{j_2} \cup \{w_{ij_2}\}))^{1/p}.$$

Equivalently, by raising both sides to the power p and multiplying both sides with $w_{i,\min}$, we have that

$$w_{ij_1} \Psi_p(N_{j_1}) \leq w_{ij_2} \Psi_p(N_{j_2} \cup \{w_{ij_2}\}).$$

Using the binary variables x_{ij} and y_{ij} to denote whether job i is assigned to machine j in the assignment N ($x_{ij} = 1$) and O ($y_{ij} = 1$) or not ($x_{ij} = 0$ and $y_{ij} = 0$, respectively), we can express the last inequality as follows:

$$\sum_j x_{ij} w_{ij} \Psi_p(N_j) \leq \sum_j y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\})$$

By summing over all jobs, we have

$$\sum_i \sum_j x_{ij} w_{ij} \Psi_p(N_j) \leq \sum_i \sum_j y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\})$$

By exchanging the double sums and since $\sum_i x_{ij} w_{ij} = L(N_j)$, we obtain

$$\sum_j L(N_j) \Psi_p(N_j) \leq \sum_j \sum_i y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\}) \quad (6)$$

We now work with the potential of assignment N . We have

$$\begin{aligned} 2\Phi(N) &= \Phi(N) + \sum_j \Psi_{p+1}(N_j) \\ &\leq \Phi(N) + (p+1) \sum_j L(N_j) \Psi_p(N_j) \\ &\leq \Phi(N) + (p+1) \sum_j \sum_i y_{ij} w_{ij} \Psi_p(N_j \cup \{w_{ij}\}) \\ &= \Phi(N) + (p+1) \sum_j \sum_i y_{ij} w_{ij} \sum_{t=0}^p \frac{p!}{(p-t)!} \Psi_{p-t}(N_j) w_{ij}^t \\ &= \Phi(N) + \sum_j \sum_{t=0}^p \frac{(p+1)!}{(p-t)!} \Psi_{p-t}(N_j) \sum_i y_{ij} w_{ij}^{t+1} \end{aligned}$$

$$\begin{aligned}
&\leq \Phi(N) + \sum_j \sum_{t=0}^p \frac{(p+1)!}{(p-t)!(t+1)!} \Psi_{p-t}(N_j) \Psi_{t+1}(O_j) \\
&= \Phi(N) + \sum_j \sum_{t=1}^{p+1} \binom{p+1}{t} \Psi_{p+1-t}(N_j) \Psi_t(O_j) \\
&\leq \Phi(N) + \sum_j \sum_{t=1}^{p+1} \binom{p+1}{t} \Psi_{p+1}(N_j)^{\frac{p+1-t}{p+1}} \Psi_{p+1}(O_j)^{\frac{t}{p+1}} \\
&= \Phi(N) + \sum_j \left(\left(\Psi_{p+1}(N_j)^{\frac{1}{p+1}} + \Psi_{p+1}(O_j)^{\frac{1}{p+1}} \right)^{p+1} - \Psi_{p+1}(N_j) \right) \\
&= \Phi(N) + \sum_j \left(\Psi_{p+1}(N_j)^{\frac{1}{p+1}} + \Psi_{p+1}(O_j)^{\frac{1}{p+1}} \right)^{p+1} - \sum_j \Psi_{p+1}(N_j) \\
&\leq \left(\left(\sum_j \Psi_{p+1}(N_j) \right)^{\frac{1}{p+1}} + \left(\sum_j \Psi_{p+1}(O_j) \right)^{\frac{1}{p+1}} \right)^{p+1} \\
&= \left((\Phi(N))^{\frac{1}{p+1}} + (\Phi(O))^{\frac{1}{p+1}} \right)^{p+1}
\end{aligned}$$

The first inequality follows by Lemma 12e, the second inequality follows by inequality (6), the second equality follows by Lemma 12c, the third equality follows by exchanging the sums, the third inequality follows since the jobs i assigned to machine j are those for which $y_{ij} = 1$ and by the definition of function Ψ_{t+1} which yields that $\Psi_{t+1}(O_j) \geq (t+1)! \sum_i y_{ij} w_{ij}^{t+1}$, the fourth equality follows by updating the limits of the sum over t , the fourth inequality follows by Lemma 12b, the fifth equality follows by the binomial theorem, the sixth equality is obvious, the fifth inequality follows by Minkowski inequality (Lemma 2) and by the definition of the potential $\Phi(N)$, and the last equality follows by the definition of the potentials $\Phi(N)$ and $\Phi(O)$.

By the above inequality, we obtain that

$$(\Phi(N))^{\frac{1}{p+1}} \leq \frac{1}{2^{\frac{1}{p+1}} - 1} (\Phi(O))^{\frac{1}{p+1}} \leq \frac{p+1}{\ln 2} (\Phi(O))^{\frac{1}{p+1}}$$

where the last inequality follows using the inequality $e^z \geq z + 1$. ■

We are now ready to bound the price of anarchy.

Theorem 17 *The price of anarchy of the game induced by the coordination mechanism CCOORD with $p = \Theta(\log m)$ is $O(\log^2 m)$. Also, for every constant $\epsilon \in (0, 1/2]$, the price of anarchy of the game induced by the coordination mechanism CCOORD with $p = 1/\epsilon - 1$ is $O(m^\epsilon)$.*

Proof. Consider a pure Nash equilibrium N and let O be the optimal assignment. Using Lemma 16, we have that $(\Phi(N))^{\frac{1}{p+1}} \leq \frac{p+1}{\ln 2} (\Phi(O))^{\frac{1}{p+1}}$. Hence, by Lemma 14, we obtain that the maximum completion time in N is at most $\frac{(p+1)^2}{\ln 2} m^{\frac{1}{p+1}} + p$ times the makespan of O . By setting $p = \Theta(\log m)$ and $p = 1/\epsilon - 1$, respectively, the theorem follows. ■

6 Discussion and open problems

Our focus in the current paper has been on pure Nash equilibria. It is also interesting to generalize the bounds on the price of anarchy of the games induced by our coordination mechanisms for mixed Nash equilibria. Recently, Roughgarden [47] defined general smoothness arguments that can be used to bound the price of anarchy of games having particular properties. Bounds on the price of anarchy over pure Nash equilibria that are proved using smoothness arguments immediately imply that the same bounds on the price of anarchy hold for mixed Nash equilibria as well. We remark that the arguments used in the current paper in order to prove our upper bounds are not smoothness arguments. At least in the case of the coordination mechanism **BCOORD**, smoothness arguments cannot be used to prove a bound on the price of anarchy as small as $O\left(\frac{\log m}{\log \log m}\right)$ since the price of anarchy over mixed Nash equilibria is provably higher in the case. We demonstrate this using the following construction. Czumaj and Voecking [18] present a game induced by the **Makespan** policy on related machines which has price of anarchy over mixed Nash equilibria at least $\Omega\left(\frac{\log m}{\log \log m}\right)$. The instance used in [18] consists of n jobs and m machines. Each machine j has a speed $\alpha_j \geq 1$ with $\alpha_1 = 1$ and each job i has a weight w_i . The processing time of job i on machine j is $w_{ij} = \alpha_j w_i$ (i.e., the inefficiencies of the jobs are the same on the same machine). Now, consider the game induced by the coordination mechanism **BCOORD** for the instance that consists of the same machines and jobs in which the processing time of job i on machine j is defined by $w'_{ij} = \alpha_j^{\frac{p}{p+1}} w_i$, i.e., the inefficiency of any job on machine j is $\alpha_j^{\frac{p}{p+1}}$. Here, p is the parameter used by **BCOORD**. By the definition of **BCOORD**, we can easily see that the game induced is identical with the game induced by **Makespan** on the original instance of [18]. Also note that, in our instance, the processing time of the jobs is not increased (and, hence, the optimal makespan is not larger than that in the original instance of [18]). Hence, the lower bound of [18] implies a lower bound on the price of anarchy over mixed Nash equilibria of the game induced by the coordination mechanism **BCOORD**.

Our work reveals several other interesting questions. First of all, it leaves open the question of whether coordination mechanisms with constant approximation ratio exist. In particular, is there any coordination mechanism that handles anonymous jobs, guarantees that the induced game has pure Nash equilibria, and has constant price of anarchy? Based on the lower bounds of [8, 25], such a coordination mechanism (if it exists) must use preemption. Alternatively, is the case of anonymous jobs provably more difficult than the case where jobs have IDs? Investigating the limits of non-preemptive mechanisms is still interesting. Notice that **AJM-1** is the only non-preemptive coordination mechanism that has approximation ratio $o(m)$ but it does not guarantee that the induced game has pure Nash equilibria; furthermore, the only known non-preemptive coordination mechanism that induces a potential game with bounded price of anarchy is **ShortestFirst**. So, is there any non-preemptive (deterministic or randomized) coordination mechanism that is simultaneously $o(m)$ -approximate and induces a potential game? We also remark that Theorem 10 does not necessarily exclude a game induced by the coordination mechanism **BCOORD** from having pure Nash equilibria. Observe that the examples in the proof of Theorem 10 do not consist of best-response moves and, hence, it is interesting to investigate whether best-response moves converge to pure Nash equilibria in such games.

Furthermore, we believe that the games induced by the coordination mechanism **CCOORD** are of independent interest. We have proved that these games belong to the class **PLS** [30]. Furthermore, the result of Monderer and Shapley [40] and the proof of Lemma 13 essentially show that each of these games is isomorphic to a congestion game. However, they have a beautiful definition as games on parallel machines that gives them a particular structure. What is the complexity of computing pure Nash equilibria in such

games? Even in case that these games are PLS-complete (informally, this would mean that computing a pure Nash equilibrium is as hard as finding any object whose existence is guaranteed by a potential function argument) like several variations of congestion games that were considered recently [1, 21, 49], it is still interesting to study the convergence time to efficient assignments. A series of recent papers [6, 15, 22, 39] consider adversarial rounds of best-response moves in potential games so that each player is given at least one chance to play in each round (this is essentially our assumption in Lemma 5 for the coordination mechanism ACOORD). Does the game induced by the coordination mechanism CCOORD converges to efficient assignments after a polynomial number of adversarial rounds of best-response moves? Although it is a potential game, it does not have the particular properties considered in [6] and, hence, proving such a statement probably requires different techniques.

Finally, recall that we have considered the maximum completion time as the measure of the efficiency of schedules. Other measures such as the weighted sum of completion times that is recently studied in [16] are interesting as well. Of course, considering the application of coordination mechanisms to settings different than scheduling is an important research direction.

Acknowledgments. I would like to thank Chien-Chung Huang for helpful comments on an earlier draft of the paper.

References

- [1] H. Ackermann, H. Röglin, and B. Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 55(6), 2008.
- [2] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS 3884, Springer, pp. 218-229, 2006.
- [3] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4), pp. 1602-1623, 2008.
- [4] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3), pp. 486-504, 1997.
- [5] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 57-66, 2005.
- [6] B. Awerbuch, Y. Azar, A. Epstein, V. S. Mirrokni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC)*, pp. 264-273, 2008.
- [7] B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P. Krishnan, and J. S. Vitter. Load balancing in the L_p norm. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 383-391, 1995.
- [8] Y. Azar, K. Jain, and V. S. Mirrokni. (Almost) optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings of the 19th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA)*, pp. 323-332, 2008.
- [9] Y. Azar, J. Naor, and R. Rom. The competitiveness of on-line assignments. *Journal of Algorithms*, 18(2), pp. 221-237, 1995.
- [10] I. Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the 19th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA)*, pp. 972-981, 2008.

- [11] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Taxes for linear atomic congestion games. *ACM Transactions on Algorithms*, 7(1), art. 13, 2010.
- [12] H.-L. Chen, T. Roughgarden, and G. Valiant. Designing network protocols for good equilibria. *SIAM Journal on Computing*, 39(5), pp. 1799-1832, 2010.
- [13] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 67-73, 2005.
- [14] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *Theoretical Computer Science*, 410(36), pp. 3327-3336, 2009.
- [15] G. Christodoulou, V. S. Mirrokni, and A. Sidiropoulos. Convergence and approximation in potential games. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS 3884, Springer, pp. 349-360, 2006.
- [16] R. Cole, J. R. Correa, V. Gkatzelis, V. Mirrokni, and N. Olver. Inner product spaces for MinSum coordination mechanisms. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, pp. 539-548, 2011.
- [17] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *Journal of Computer and System Sciences*, 72(3), pp. 444-467, 2006.
- [18] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. *ACM Transactions on Algorithms*, 3(1), 2007.
- [19] C. Dürr and T. Nguyen Kim. Personal communication, 2009.
- [20] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibria in load balancing. *ACM Transactions on Algorithms*, 3(3), 2007.
- [21] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 604-612, 2004.
- [22] A. Fanelli, M. Flammini, and L. Moscardelli. Speed of convergence in congestion games under best response dynamics. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, LNCS 5125, Springer, pp. 796-807, 2008.
- [23] M. Feldman, M. Rosenthal, and O. Sivan. Makespan minimization through coordination mechanisms. Manuscript, 2010.
- [24] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 277-285, 2004.
- [25] L. Fleischer and Z. Svitkina. Preference-constrained oriented matchings. In *Proceedings of the 7th Workshop on Analytic Algorithmic and Combinatorics (ANALCO)*, pp. 66-73, 2010.
- [26] D. Fotakis, S. Kontogiannis, and P. Spirakis. Selfish unsplittable flows. *Theoretical Computer Science*, 340(3), pp. 514-538, 2005.
- [27] G. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. 2nd edition, Cambridge University Press, 1952.
- [28] O. H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM*, 24(2), pp. 280-289, 1977.
- [29] N. Immorlica, L. Li, V. S. Mirrokni, and A. Schulz. Coordination mechanisms for selfish scheduling. *Theoretical Computer Science*, 410(17), pp. 1589-1598, 2009.
- [30] D. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37, pp. 79-100, 1988.

- [31] A. C. Kaporis and P. G. Spirakis. The price of optimum in Stackelberg games on arbitrary single commodity networks and latency functions. *Theoretical Computer Science*, 410(8-10), pp. 745-755, 2009.
- [32] G. Karakostas, and S. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 268-276, 2004.
- [33] Y. A. Korilis, A. A. Lazar, and A. Orda. Architecting noncooperative networks. *IEEE Journal on Selected Areas in Communications*, 13(7), pp. 1241-1251, 1995.
- [34] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1), pp. 161-173, 1997.
- [35] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS 1563, Springer, pp. 404-413, 1999.
- [36] V. S. Anil Kumar and M. V. Marathe. Improved results for Stackelberg scheduling strategies. In *Proceedings of the 29th Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, LNCS 2380, Springer, pp. 776-787, 2002.
- [37] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46, pp. 259-271, 1990.
- [38] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13, pp. 111-124, 1996.
- [39] V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. In *Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, LNCS 3122, Springer, pp. 183-194, 2004.
- [40] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14: 124-143, 1996.
- [41] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.
- [42] C. H. Papadimitriou. Algorithms, games and the internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 749-753, 2001.
- [43] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2, pp. 65-67, 1973.
- [44] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2), pp. 332-350, 2004.
- [45] T. Roughgarden. On the severity of Braess's Paradox: Designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5), pp. 922-953, 2006.
- [46] T. Roughgarden. Routing games. In [41], Chapter 18, pp. 461-486, 2007.
- [47] T. Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 513-522, 2009.
- [48] P. Schuurman and T. Vredeveld. Performance guarantees of local search for multiprocessor scheduling. *INFORMS Journal on Computing*, 19(1), pp. 52-63, 2007.
- [49] A. Skopalik and B. Vöcking. Inapproximability of pure Nash equilibria. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 355-364, 2008.
- [50] S. Suri, C. D. Toth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 47(1), pp. 77-96, 2007.
- [51] C. Swamy. The effectiveness of Stackelberg strategies and tolls for network congestion games. In *Proceedings of the 18th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1133-1142, 2007.
- [52] B. Vöcking. Selfish load balancing. In [41], Chapter 20, pp. 517-542, 2007.

A Proof of Lemma 12

The properties clearly hold if A is empty or $k = 1$. In the following, we assume that $k \geq 2$ and $A = \{a_1, \dots, a_n\}$ for integer $n \geq 1$.

a. Clearly,

$$L(A)^k = \left(\sum_{t=1}^k a_t \right)^k = \sum_{1 \leq d_1 \leq \dots \leq d_k \leq n} \zeta(d_1, \dots, d_k) \prod_{t=1}^k a_{d_t}$$

where $\zeta(d_1, \dots, d_k)$ are multinomial coefficients on k and, hence, belong $\{1, \dots, k!\}$. The property then follows by the definition of $\Psi_k(A)$.

b. We can express $\Psi_{k-1}(A)^k$ and $\Psi_k(A)^{k-1}$ as follows:

$$\begin{aligned} \Psi_{k-1}(A)^k &= ((k-1)!)^k \left(\sum_{1 \leq d_1 \leq \dots \leq d_{k-1} \leq n} \prod_{t=1}^k a_{d_t} \right)^k \\ &= ((k-1)!)^k \sum_{1 \leq d_1 \leq \dots \leq d_{k(k-1)} \leq n} \zeta_1(d_1, \dots, d_{k(k-1)}) \prod_{t=1}^{k(k-1)} a_{d_t}. \\ \Psi_k(A)^{k-1} &= (k!)^{k-1} \left(\sum_{1 \leq d_1 \leq \dots \leq d_k \leq n} \prod_{t=1}^k a_{d_t} \right)^{k-1} \\ &= (k!)^{k-1} \sum_{1 \leq d_1 \leq \dots \leq d_{k(k-1)} \leq n} \zeta_2(d_1, \dots, d_{k(k-1)}) \prod_{t=1}^{k(k-1)} a_{d_t}. \end{aligned}$$

So, both $\Psi_{k-1}(A)^k$ and $\Psi_k(A)^{k-1}$ are sums of all monomials of degree $k(k-1)$ over the elements of A with different coefficients. The coefficient $\zeta_1(d_1, \dots, d_{k(k-1)})$ is the number of different ways to partition the multiset $D = \{d_1, \dots, d_{k(k-1)}\}$ of size $k(k-1)$ into k disjoint ordered multisets each of size $k-1$ so that the union of the ordered multisets yields the original multiset. We refer to these partitions as $(k, k-1)$ -partitions. The coefficient $\zeta_2(d_1, \dots, d_{k(k-1)})$ is the number of different ways to partition D into $k-1$ disjoint ordered multisets each of size k (resp. $(k-1, k)$ -partitions). Hence, in order to prove the property, it suffices to show that for any multiset $\{d_1, \dots, d_{k(k-1)}\}$,

$$\frac{\zeta_1(d_1, \dots, d_{k(k-1)})}{\zeta_2(d_1, \dots, d_{k(k-1)})} \leq \frac{k^{k-1}}{(k-1)!}. \quad (7)$$

Assume that some element of D has multiplicity more than one and consider the new multiset $D' = \{d_1, \dots, d'_i, \dots, d_{k(k-1)}\}$ that replaces one appearance d_i of this element with a new element d'_i different than all elements in D . Then, in order to generate all $(k, k-1)$ -partitions of D' , it suffices to consider the $(k, k-1)$ -partitions of D and, for each of them, replace d_i with d'_i once for each of the ordered sets in which d_i appears. Similarly, we can generate all $(k-1, k)$ -partitions of D' using the $(k-1, k)$ -partitions of D .

Since the number of sets in $(k, k-1)$ -partitions is larger than the number of sets in $(k-1, k)$ -partitions, we will have that

$$\frac{\zeta_1(d_1, \dots, d'_i, \dots, d_{k(k-1)})}{\zeta_2(d_1, \dots, d'_i, \dots, d_{k(k-1)})} \geq \frac{\zeta_1(d_1, \dots, d_{k(k-1)})}{\zeta_2(d_1, \dots, d_{k(k-1)})}.$$

By repeating this argument, we obtain that the ratio at the left-hand side of inequality (7) is maximized when all d_i 's are distinct. In this case, both ζ_1 and ζ_2 are given by the multinomial coefficients

$$\zeta_1(d_1, \dots, d_{k(k-1)}) = \left(\underbrace{k-1, \dots, k-1}_{k \text{ times}} \right) = \frac{(k(k-1))!}{((k-1)!)^k}$$

and

$$\zeta_2(d_1, \dots, d_{k(k-1)}) = \left(\underbrace{k, \dots, k}_{k-1 \text{ times}} \right) = \frac{(k(k-1))!}{(k!)^{k-1}}$$

and their ratio is exactly the one at the right-hand side of the inequality (7).

c. The property follows easily by the definition of function Ψ_k by observing that all the monomials of degree k over the elements of A that contain b^t are generated by multiplying b^t with the terms of $\Psi_{k-t}(A)$.

d. By property (c), we have

$$\begin{aligned} \Psi_k(A \cup \{b\}) - \Psi_k(A) &= \sum_{t=1}^k \frac{k!}{(k-t)!} b^t \Psi_{k-t}(A) \\ &= kb \sum_{t=1}^k \frac{(k-1)!}{(k-t)!} b^{t-1} \Psi_{k-t}(A) \\ &= kb \sum_{t=0}^{k-1} \frac{(k-1)!}{(k-1-t)!} b^t \Psi_{k-1-t}(A) \\ &= kb \Psi_{k-1}(A \cup \{b\}). \end{aligned}$$

e. Working with the right-hand side of the inequality and using the definitions of L and Ψ_{k-1} , we have

$$\begin{aligned} kL(A)\Psi_{k-1}(A) &= k! \left(\sum_{t=1}^n a_t \right) \cdot \sum_{1 \leq d_1 \leq \dots \leq d_{k-1} \leq n} \prod_{t=1}^{k-1} a_{d_t} \\ &\geq k! \sum_{1 \leq d_1 \leq \dots \leq d_k \leq n} \prod_{t=1}^k a_{d_t} \\ &= \Psi_k(A). \end{aligned}$$

The equalities follow obviously by the definitions. To see why the inequality holds, observe that the multiplication of the sum of all monomials of degree 1 with the sum of all monomials of degree $k-1$ will be a sum of all monomials of degree k , each having coefficient at least 1. ■

f. The proof follows by the derivation below in which we use property (c), the fact that $\Psi_t(\{b\}) = t!b^t$ by the definition of function Ψ_t , property (b), and the binomial theorem. We have

$$\begin{aligned}
\Psi_k(A \cup \{b\}) &= \sum_{t=0}^k \frac{k!}{(k-t)!} b^t \Psi_{k-t}(A) \\
&= \sum_{t=0}^k \binom{k}{t} \Psi_{k-t}(A) \Psi_t(\{b\}) \\
&\leq \sum_{t=0}^k \binom{k}{t} \Psi_k(A)^{\frac{k-t}{k}} \Psi_k(\{b\})^{\frac{t}{k}} \\
&= \left(\Psi_k(A)^{1/k} + \Psi_k(\{b\})^{1/k} \right)^k .
\end{aligned}$$

■